Image-Based-Retrieval system using FAISS Algorithm and SigLIP model

Senthil Thanneermalai[*]
st108@rice.edu

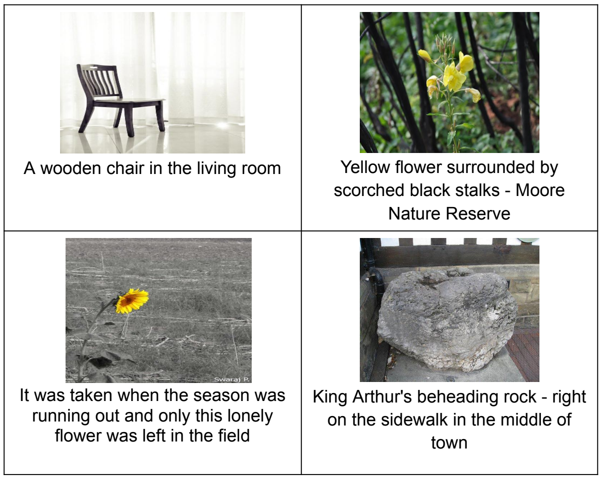
Kaushal Kumar Agarwal[*]
ka62@rice.edu

Abstract

Image retrieval systems are becoming popular daily as the demand for image search increases. Since images can be represented as feature vectors, they can be easily stored in the vector database for faster query retrieval. This paper focuses on extracting image features from the SBU Captions Dataset [5] using a pre-trained SigLIP (Sigmoid Loss for Language Image Pre-Training) model and evaluating the FAISS (Facebook AI Similarity Search) library for retrieving k similar images for a given input image. The final results and performance of the image retrieval system are assessed using different metrics, including Recall@K and system latency. The system's latency is measured using brute force Euclidean distance and various FAISS indexes.

1. Introduction

In today's world, efficiently searching and retrieving information from large datasets, primarily containing images, is crucial. This report focuses on image retrieval systems, exploring how images can be represented as feature vectors or matrices using the SigLIP model for efficient storage and retrieval. The process of finding a similar image involves multiple vector comparisons, underscoring the importance of an efficient retrieval mechanism for these vectors. To address this issue, the Facebook AI Similarity Search (FAISS) library was utilized. FAISS, specifically designed for the efficient similarity search and clustering of dense vectors, has gained popularity due to its ability to find similar vectors among large datasets, in this case, a dataset of 1 million images. A key component of FAISS is its IndexIVFFlat; this clustering-based technique is used to index and retrieve vectors efficiently. The results were evaluated using the Recall at K metric, and the time taken to retrieve similar images from a dataset of 1 million was compared using different strategies.



A wooden chair in the living room

Yellow flower surrounded by scorched black stalks - Moore Nature Reserve

It was taken when the season was running out and only this lonely flower was left in the field

King Arthur's beheading rock - right on the sidewalk in the middle of town

Figure 1. Here we show some sample images from the SBU Captions Dataset that we use in our work. The objective is to retrieve k images most similar to the given input image.

2. Related Work

Previously, performance analysis of FAISS [12] has been carried out on high-dimensional features of images extracted using the VGG16 CNN model. Moreover, the datasets (UKBench, Oxford5k, and Paris6k) chosen contain only up to 10,000 maximum photos. The experiments confirmed that FAISS could significantly reduce the latency for querying similar images, and it performed best with the concatenation of the HSV descriptor and the feature vector obtained from VGG16.

Another research work [13] for faster retrieval of an image from the database includes an algorithm called Feature Levels combined with the Database Revision (DR) technique. The novel approach of extracting and categorizing image features into different levels improved the search algorithm compared to the conventional method.

Authors in [14] have built a content-based image retrieval system specifically for medical images and documents. They use multiple foundational models such as Vision Transformer (ViT), CLIP, and MedCLIP to extract feature vectors and store them in a vector database. For retrieval, they calculated the cosine similarity between every

[*] Department of Computer Science, Rice University

vector and sorted them in descending order such that the most similar image vector is first in the list. This leads to a significantly higher latency and more memory for storing these vectors.

In the implementation, SigLIP was utilized for feature extraction. The similarity search employed in this system demonstrated enhanced speed and efficiency, both in terms of space utilization and accuracy.

3. Model

As shown in Figure 2, a multimodal and crossmodal neural network known as the SigLIP model [15], a variant of the well-established CLIP model designed to extract the image features in fixed vector dimensions, has been utilized. The SigLIP model distinguishes itself by replacing the CLIP's (Contrastive Language-Image Pre-Training) contrastive loss function with a pairwise Sigmoid Loss function, simplifying the training process and enhancing the model's ability to learn the context of text in relation to images. SigLIP training is similar to the CLIP model [4], where over 500 million pairs of captions and images were used to train the model such that the model learns the context of the text associated with an image rather than directly predicting the image from text or vice versa. The model was trained using 4 TPUv4 chips, demonstrating its efficiency by achieving an accuracy of 84.5% on ImageNet zero-shot learning.

Two variants of the SigLIP model have been used: google/siglip-base-patch16-224 [7] and google/siglip-so400m-patch14-384 [8]. The former model outputs a feature vector of 768 dimensions, while the latter produces a feature vector of 1152 dimensions. The SigLIP base model aims to enhance zero-shot classification accuracy on ImageNet by employing a simple pairwise sigmoid loss instead of the standard softmax normalization used in contrastive learning. This approach allows for the scaling of batch sizes and yields better performance even with smaller batches. The shape-optimized SigLIP model, pre-trained on WebLi [9], incorporates the SoViT-400m architecture [11], optimizing the model for high-resolution inputs and providing a more robust framework for tasks such as zero-shot image classification and image-text retrieval.

Images from the SBU Captions Dataset are processed using a dataset loader to extract image features and passed through the SigLIP model in batches of 8. The preprocessing pipeline includes resizing the images to a uniform size of 256x256 pixels, converting them to tensors, and normalizing them using mean and standard deviation functions to ensure optimal preparation for feature extraction by each variant of the SigLIP model. This standardized approach allows for consistent and accurate evaluation of the Recall at K metric.

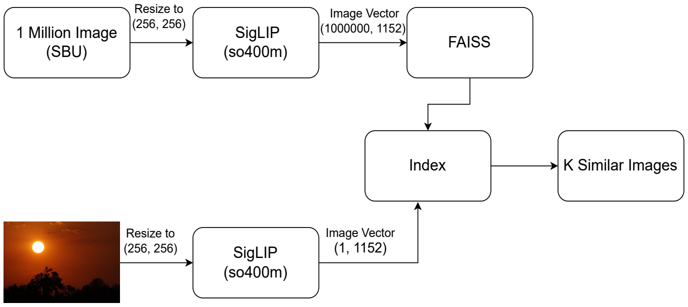


Figure 2. Here, we show a figure detailing our model components. The input is an image. The output is a list of indexes of K-predicted similar images.

FAISS (Facebook AI Similarity Search) is a library developed by Facebook that enables rapid searching of k similar images from a very large dataset [3]. In the described implementation, indexes are created on previously extracted image vectors using FAISS. This library preprocesses the vectors and indexes them to facilitate efficient similarity searches. It has the following types of Indexing available for use [2]

1. **IndexFlatL2**: This type of FAISS index uses euclidean distance L2 between every vector to calculate the vector similarities. This method is also known as K Nearest Neighbor Search (KNN).

2. **IndexIVFFlat**: This divides the vector into multiple cells and searches for similar vectors only in specific cells. This is also known as Approximate Nearest Neighbor (ANN). This leads to a reduction in accuracy but increases the vector search considerably.

3. **IndexHNSWFlat**: This index type is also an ANN that organizes the vectors in a Hierarchical graph-based manner, allowing for faster search.

4. **IndexLSH**: This index uses Hashing methods to map similar items in the same buckets to perform an approximate nearest neighbor search.

In the current implementation, the IndexFlatL2 and IndexIVFFlat indices are utilized to calculate the similarity among vectors extracted using the SigLIP model. After experimentation with different values of Voronoi cells for the IndexIVFFlat, it was determined that the optimum number of cells is 50. The results from FAISS are ranked in descending order, with the most similar images presented first, thus eliminating the need for manual ranking of the images. Additionally, two versions of the SigLIP model are compared in terms of the latency involved in retrieving k-similar images and the Recall@K metric.

4. Experiments and Results

Initially, the SigLIP model with version siglip-base-patch16-224 was selected to extract feature vectors from the 1 million images in the SBU Captions Dataset. Subsequently, a FlatL2 index was constructed using FAISS, employing a brute force approach, to retrieve k nearest images. To assess the system's recommendations, the Recall@K metric was employed. Recall@K [10] measures the proportion of correctly identified relevant items in the top K recommendations out of the total number of relevant items in the dataset, as defined in Equation 1.

$$\text{Recall@K} = \frac{\text{Number of relevant items @ K}}{\text{Total number of relevant items}} \tag{1}$$

One challenge in calculating the Recall@K metric is ascertaining the ground truth values among the recommended images. When utilizing a query image not present in the dataset, it becomes impossible to directly evaluate the system's recommendations without manually inspecting the images to determine their similarity to the query image.

As shown in Figure 3, one method to overcome this challenge involves selecting an image and its corresponding caption from the dataset. The caption is then tokenized and transformed into a feature vector. This vector serves as the basis for retrieving similar images using the FAISS image vector index previously constructed based on the SBU Captions dataset. Subsequently, the rank at which the original query image is located within the set of recommended images is determined. Utilizing this value, the Recall@K metric was computed for a subset of 1000 random images to gauge the system's accuracy in finding similar images. It was observed that the Recall@K for k = 1, 10, 50, 100 is 0 for siglip-base-patch16-224.

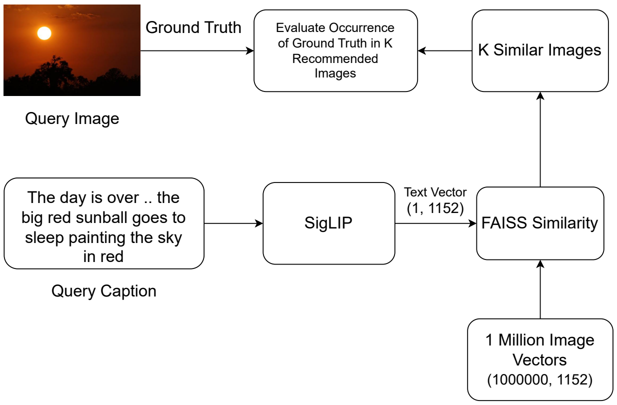

Ground Truth
Evaluate Occurrence of Ground Truth in K Recommended Images
K Similar Images
Query Image
The day is over .. the big red sunball goes to sleep painting the sky in red
Query Caption
SigLIP
Text Vector (1, 1152)
FAISS Similarity
1 Million Image Vectors (1000000, 1152)

Figure 3. Evaluation of Recommended images using Recall@K metric for 1000 images

This prompted the selection of a newer, more refined model named siglip-so400m-patch14-384. With this model, features were extracted for 1 million images from the SBU Captions Dataset, and FAISS's FlatL2 and IVFFlat indexes were utilized. As indicated in Table 1, the model effectively predicted the correct query images using solely the caption vector and FAISS indexes.

Table 1. Recall@K Analysis for 1000 Images in 1M Dataset Using FAISS Indexes

Index Name	**Recall@1**	**Recall@10**	**Recall@50**	**Recall@100**
IndexFlatL2 (%)	8.80	35.30	67.40	87.00
IndexIVFFlat (%)	0.70	1.60	3.40	4.10

From the above metric values, it can be concluded that the system performs well in recommending the query image using the IndexFlatL2 FAISS index. It is observed that the query image is present in 87% of the cases when considering the top 100 recommended images. Similarly, the query image is present in 35% of the cases when considering the top 10 recommended images. However, in the case of IndexIVFFlat, despite satisfactory results for new query images, the Recall@K metric does not show comparable performance. To address this issue, further exploration of other FAISS indexes, such as IndexHNSWFlat and IndexLSH, is planned for future work.

Experiments were conducted using the input image shown in Figure 4 to measure latency for searching k-similar images within an index of 1 million image vectors extracted from the SBU Captions Dataset. The metrics shown in Table 2 and Table 3 highlight the efficiency and speed of using Faiss's indexes in our image retrieval system.

Figure 4. Sample image from the Flickr 8K Dataset [1]. This image is used as an input to evaluate the retrieval times of K-similar images from an index of 1M image vectors of the SBU Captions Dataset.

Table 2. Comparative Analysis of Time Taken to Retrieve K Similar Images from an Index of 1 Million Images using SigLIP base Model

Index Name	K=10	K=100	K=1000	K=10000	K=100000
Without FAISS (Time in sec)	44.1362	55.8243	67.0342	61.0510	48.6658
IndexFlatL2 (Time in sec)	0.5619	0.5624	0.5677	0.5618	0.5879
IndexIVFFlat (Time in sec)	0.0219	0.0204	0.0194	0.0190	0.0236

Table 3. Comparative Analysis of Time Taken to Retrieve K Similar Images from an Index of 1 Million Images SigLIP so400m Model

Index Name	K=10	K=100	K=1000	K=10000	K=100000
Without FAISS (Time in sec)	116.0751	125.7797	121.3240	131.2460	123.6157
IndexFlatL2 (Time in sec)	0.4266	0.3330	0.3784	0.4812	0.5135
IndexIVFFlat (Time in sec)	0.0694	0.0596	0.0132	0.0573	0.0648

For the SigLIP base model, it was observed that the brute force approach, which involves comparing each vector with every other vector by calculating Euclidean distance, took more than a minute as the value of K increased from 10 to 1000. However, employing FAISS's FlatL2 and IVFFlat indexes led to a significant reduction in the latency for retrieving the same K similar images, with retrieval time dropping from a minute to half a second and even to milliseconds.

For the SigLIP shape optimized model, the time taken to retrieve similar images using the brute force approach almost doubled (around 2 minutes) for the same values of K. Despite the index size of this model being almost double that of the base model, the latency of the image retrieval system using FAISS was not significantly impacted.

Additionally, The time taken to create the indexes was also noted. For the base model, IndexFlatL2 was created in 1.46 seconds, while the IndexIVFFlat with 50 cells took slightly longer at 2.96 seconds. For the shape optimized model, these times increased to 3.19 and 5.42 seconds respectively. Despite the initial time investment required in creating the IndexIVFFlat, its enhanced retrieval speed makes it a valuable tool for large-scale image retrieval tasks.

The experiments detailed above were executed on a system equipped with a Macbook M2 chip and 16GB of RAM in the CPU. For the extraction of 1M image features using the SigLIP so400m model, a more robust setup was employed. This setup comprised of two GPUs (2 x Tesla V100) in a single node, supplemented with 100GB RAM, housed in Rice University's NOTS server [6]. The extraction process for 100K image features was completed in approximately 3 hours and 30 minutes. However, for a larger scale extraction involving 1 million images, the process extended to nearly 35 hours. This underlines the computational intensity of large-scale feature extraction tasks.

In terms of project contribution, image features were extracted using the SigLIP base model (siglip-base-patch16-224), and the FAISS similarity index was evaluated using the Recall@K metric by one member of the team. Concurrently, another member focused on extracting image features using a newer model (siglip-so400m-patch14-384) and calculating the latency metric.

5. Conclusion

In this project, two different FAISS indexes were constructed on 1 million image vectors extracted from the SBU Captions Dataset using two distinct versions of the SigLIP model. The performance of these indexes was evaluated using various metrics. It was observed that both indexes yielded satisfactory results with the query images. However, regarding evaluation metrics, the IndexFlatL2 demonstrated superior Recall@K values compared to IndexIVFFlat. Conversely, IndexIVFFlat exhibited faster retrieval times for K similar images than IndexFlatL2. Experimentation with IndexHNSWFlat and IndexLSH indexes from FAISS is planned for future work.

References

[1] A. Jain, adityajn105, "Flickr 8K dataset," Kaggle, https://www.kaggle.com/datasets/adityajn105/flickr8k (accessed Apr. 22, 2024).

[2] Facebookresearch, "Facebookresearch/faiss: A library for efficient similarity search and clustering of dense vectors.," GitHub, https://github.com/facebookresearch/faiss (accessed Apr. 22, 2024).

[3] H. Jegou, M. Douze, and J. Johnson, "FAISS: A library for efficient similarity search," Engineering at Meta, https://engineering.fb.com/2017/03/29/data-infrastructure/faiss-a-library-for-efficient-similarity-search/ (accessed apr. 22, 2024).

[4] N. Kafritsas, "Clip: The most influential AI model from openai- and how to use it," Medium, https://towardsdatascience.com/clip-the-most-influential-ai-model-from-openai-and-how-to-use-it-f8ee408958b1 (accessed Apr. 22, 2024).

[5] Papers with code - SBU Captions Dataset," Dataset — Papers With Code, https://paperswithcode.com/dataset/sbu-captions-dataset (accessed Apr. 22, 2024).

[6] "CRC Getting Started on NOTS," Rice.edu, 2020. https://kb.rice.edu/108237 (accessed Apr. 23, 2024).

[7] "Google/Siglip-Base-Patch16-224 · Hugging Face," Huggingface.co, 9 Apr. 2024, huggingface.co/google/siglip-base-patch16-224. Accessed 21 Apr. 2024.

[8] "Google/Siglip-So400m-Patch14-384 · Hugging Face." Huggingface.co, 9 Apr. 2024, huggingface.co/google/siglip-so400m-patch14-384. Accessed 21 Apr. 2024.

[9] “Papers with Code - Webli Dataset.” WebLI Dataset — Papers With Code, paperswithcode.com/dataset/webli. Accessed 23 Apr. 2024.

[10] “Precision and recall at K in ranking and recommendations,” Evidently AI - Open-Source ML Monitoring and Observability, https://www.evidentlyai.com/ranking-metrics/precision-recall-at-k: :text=Precision

[11] I. Alabdulmohsin, X. Zhai, A. Kolesnikov, and L. Beyer. Getting vit in shape: Scaling laws for compute-optimal model design, Jan 2024.

[12] A. Gupta, D. Agarwal, Veenu, and M. P. S. Bhatia. Performance analysis of content based image retrieval systems. In *2018 International Conference on Computing, Power and Communication Technologies (GUCON)*, pages 899–902, 2018.

[13] S. Sreedevi and S. Sebastian. Fast image retrieval with feature levels. In *2013 Annual International Conference on Emerging Research Areas and 2013 International Conference on Microelectronics, Communications and Renewable Energy*, pages 1–4, 2013.

[14] D. B. M. B. S. X. L. K. P. S. T. P. P. F. J. K. M.-H. Stefan Denner, David Zimmerer. Leveraging foundational models for content based image retrieval sytem in radiology. In *Computer Vision and Pattern Recognition*, 2024.

[15] A. K. L. B. Xiaohua Zhai, Basil Mustafa. Sigmoid loss for language image pre-training. In *Proceedings of the International Conference of Computer Vision*, 2023.